

WHAT IS CLAIMED IS:

1. (Original) A client machine for remotely executing application functionality, the client machine communicatively coupled to a server, a forms manager residing on the server, the client machine comprising:

AH a view manager, the view manager operable for generating a graphical user interface environment, the graphical user interface environment comprising graphical objects, the view manager manipulating the graphical objects in response to communication from the forms manager;

instructions for characterizing events at the client machine as either requiring processing by the application server or processing by the view manager;

instructions for parsing events in an event queue at the client machine to determine when and in what order to process the events; and

instructions for extracting events that require processing by the application engine and sending the events that require processing by the application engine to the server.

2. (Original) The client machine of Claim 1, the client machine further comprising: instructions for maintaining stack synchronization between the client machine and the server.

3. (Original) The client machine of Claim 1 wherein the graphical objects comprise predefined graphical elements.

4. (Original) The client machine of Claim 1 wherein the forms manager maintains data objects on the server, the data objects being associated with the graphical objects on the client machine.

5. (Original) The client machine of Claim 1 wherein the client machine is communicatively coupled to the server through a global network.

6. (Original) The client machine of Claim 1 wherein the client machine is communicatively coupled to the server through a network using TCP/IP.

7. (Original) The client machine of Claim 1 wherein the client machine is communicatively coupled to the server through a dispatcher.

8. (Original) The client machine of Claim 1 wherein the client machine is communicatively coupled to the server through a wireless network.

AH 9. (Original) A server, the server communicatively coupled to at least one client machine, the server comprising:
a forms manager, the forms manager operable to direct a view manager residing on the at least one client machine to render a graphical object, and
at least one data object, the forms manager storing data associated with the graphical data object in the at least one data object; and
an application logic accessible by the forms manager, the forms manager applying the application logic to a message sent by the view manager associated with an event from an event queue residing on the at least one client machine, the event being characterized by the at least one client machine as requiring processing by the server.

10. (Original) The server of Claim 9, the server further comprising:
an application-programming interface, the application programming interface operable to permit access to the core functionality of the forms manager.

11. (Original) The server of Claim 9, the server further comprising:
a database interface accessible to the forms manager.

12. (Original) The server of Claim 11, the server further comprising:
a database accessible to the forms manager through the database interface.

13. (Original) The server of Claim 9 wherein the server is communicatively coupled to the at least one client machine through a global network.

14. (Original) The server of Claim 9 wherein the server is communicatively coupled to the at least one client machine through a dispatcher.

15. (Original) The server of Claim 9 wherein the server is communicatively coupled to the at least one client machine through a network using TCP/IP.

A1 16. (Original) The server of Claim 9 wherein the server is communicatively coupled to the at least one client machine through a wireless network.

17. (Original) A method for maintaining stack synchronization between a client application and a remote application engine, the method comprising:

determining whether a next event in an event queue is to be processed by the remote application engine;
selectively sending an event message associated with the next event to the remote application engine; and
selectively delaying the processing of a subsequent event until a response is received from the remote application engine.

18. (Original) The method of Claim 17, the method further comprising:
manipulating the display and access of graphical objects in response to the response received from the remote application engine.

19 (Original) The method of Claim 17 wherein the remote application engine resides on a server.

20. (Original) The method of Claim 17 wherein the remote application engine and client application are communicatively coupled through a network.